WoT Protocol Binding New Features

August 28, 2018

New patterns in Protocol Bindings

- Extended Action pattern
- Event delivery over websockets
- Event Delivery over webhooks (also Lambdas)
- Mutiplexed Event Streams
- Bulk/Batch Property Interaction
- URI Templates in Protocol Bindings

Roadmap and priorities

- September on-line plugfest
- TPAC
- Post-TPAC

Extended Action pattern

- Invoking an Action may create a new interaction instance for tracking and logging
- Protocol Binding specifies an outputMediaType

```
"forms": [
  {
    "href": "/example/light/setlevel",
    "rel": ["invokeAction"],
    "inputMediaType": "application/json",
    "outputMediaType": "application/sd+json",
    "http:methodName": "POST"
  }
]
```

```
{
  "@type": ["iot:ActionInstance"],
  "properties": {
    "status": {
      "@type": ["iot:ActionStatus"],
      "type": "object",
        "properties":{
          "currentLevel": {
            "type": "number",
            "@type": ["iot:LevelData"]
          },
          "remainingTime": {
            "type": "number",
            "@type":
["iot:RemainingTransitionTimeData", "..."]
          },
          "status": {
            "type": "string",
            "@type": ["iot:ActionStatusData"]
          }
        },
      "forms": [
        {
          "href": "/example/light/f3cca270",
          "mediaType": "application/json",
          "rel": ["readProperty", "observeProperty"]
        }
      ]
    }
  },
  "actions": {
    "cancel":{
      "@type": "CancelAction",
      "forms": [
        {
          "href": "/example/light/f3cca270",
          "rel": ["invokeAction"],
          "http:methodName": "DELETE"
        }
      ]
    }
```

Event delivery over websockets

- Simple protocol binding with ws: or wss: URI in the form
- Event payload is described in the dataSchema

Event delivery over websockets

```
{
  "events":{
    "powerstate": {
      "@type": ["Event", "iot:powerstatechange"],
      "type": "object",
      "properties": {
        "newstate": {
          "type": "string",
          "@type": ["iot:PowerStateData"]
        }
      },
      "form":[
        {
          "href": "ws://192.168.1.1:53977",
          "mediaType": "applicatio/json",
          "rel": ["subscribeEvent"]
        }
      ]
    }
```

Notification Payload

```
{"newstate":"powerstate-low-voltage"}
```

Event delivery over webhooks

- Roles are reversed and the thing pushes events to the webhook
- Client creates a webhook and provides a server URI for the thing to push events to
- Suitable for multiplexed event streams
- Webhooks have a life cycle, create to delete
- More reliable delivery due to push model
- Also could be used for thing to thing orchestration

```
"actions":{
  "createwebhook":{
    "@type": ["CreateWebHook"],
    "input":{
      "type": "object",
      "properties":{
        "sourceEvent": {
          "type": "string",
          "@type": "WebHookSource",
          "description": "the local event source: name or label"
        },
        "targetURI": {
          "type": "string",
          "@type": "WebHookDestination",
          "description": "the URI events are to be delivered to"
        },
        "targetMethod": {
          "type": "string",
          "@type": "WebHookMethod",
          "description": "transfer protocol method to use"
        }
      }
    },
    "form":[
      {
        "href": "/example/webhooks/",
        "http:methodName": "POST",
        "rel": ["invokeAction"],
        "inputMediaType": "application/json",
        "outputMediaType": "application/td+json"
```

```
"@type": ["WebHook"],
"properties":{
  "status": {
    "type": "object",
    "properties": {
      "sourceEvent": {
        "type": "string",
        "@type": "WebHookSource",
        "description": "the local event source: name or label"
      },
      "targetURI": {
        "type": "string",
        "@type": "WebHookDestination",
        "description": "the URI events are to be delivered to"
      },
      "targetMethod": {
        "type": "string",
        "@type": "WebHookMethod",
        "description": "transfer protocol method to use"
      },
      "lastError": {
        "type": "string",
        "@type": "http:errorType"
      }
    },
    "form": [
      {
        "href": "/example/webhooks/fc33af7d",
        "rel": ["readProperty", "observeProperty"]
```

```
"actions":{
  "remove": {
    "@type": ["DeleteWebHook"],
    "form": [
      {
        "href": "/example/webhooks/fc33af7d",
        "rel": ["deleteInteraction"],
        "http:methodName": "DELETE"
      }
    ]
  }
}
```

Multiplexed events using EventSource as a sub-protocol

- Multiplexing aggregates events from across time and space, multiple event types in an event stream
- EventSource provides a way to identify the sources of individual events in a multiplexed event stream
- Concept of a notification collection resource which provides the multiplexed event stream
- Events to notify are selected from available events using a filter or list
- Event collectors may be dynamically created to provide customized event streams

```
{
  "events":{
    "powerstate": {
      "label": "powerstate",
      "@type": ["Event", "iot:powerstatechange"],
      "type": "object",
      "properties": {
        "newstate": {
          "type": "string",
          "@type": ["iot:PowerStateData"]
        }
      },
      "form":[
        {
          "href": "http://192.168.1.1",
          "mediaType": "application/json",
          "http:headers":
            [
              {
                "http:fieldName": "Accept",
                "http:fieldValue": "text/event-stream"
              }
            ],
          "rel": ["subscribeEvent"],
          "subProtocol": "EventSource"
        }
      ]
    }
  }
}
```

Multiplex Notification Payload

```
event:powerstate
data:{"newstate":"powerstate-low-voltage"}
```

Event Collector Resource

- Filter capability allows selection of one or more Events to include in the Event Stream
- Collector Resource could be exposed for each Event that supports Subscription, and/or exposed for the entire TD
- Multiple Event Streams using multiple collectors
- Delivery method could be configured from a list of supported delivery methods (ws/wss, EventSource, webhook) and use EventSource sub-protocol

Bulk Property Interaction

- Selective controls vs. all properties
- Could use a collection of links to properties
- Could use a filter

URI Templates

- RFC 6570
- Substitution of path and query variables at runtime
- JSON Schema to describe the variables, including semantic annotation, serialized as URI data

URI template in the form:

```
  "href": "/example/{instance}/actuation{?level}"
```

Schema in the interaction description:

```
  "uri-properties" : {
    "level": {
      "type": "number",
      "@type": "iot:LevelData",
      "minimum": 0,
      "maximum": 254
    },
    "instance": {
(etc.)
```